Server Sizing

Performance Impact of CPU and Disk on Server Capacity

More and more companies are building distributed information infrastructures with Intel-based servers. Those who purchase, configure and deploy Intel-based servers must choose among various vendors, options and parts that will provide sufficient performance to clients. Striking the right balance among the various sub-systems and parts is critical to configuring a server. This document describes an approach to configuring a server, with a focus on how CPU capacity affects overall performance.

How do you determine the optimal CPU model and number of CPUs for a server? To date, most users make an educated guess based on intuition, rules of thumb, or vendor and magazine benchmarks. The problem with these methods is that they may not result in a server configuration that is right for your particular set of circumstances. What's missing is a picture of the server's capacity. The capacity of a server is a measure of its performance potential. An idle server does not exhibit capacity, so the basic approach to measuring computer capacity is to put the computer to work, put a load on the system, and measure the rate at which the work is accomplished.

We tested the effects of changing server configuration with single, dual and quad Intel Pentium® Pro processors to determine the impact of CPU capacity on overall server performance. We used Dynameasure*, a capacity planning and performance measurement tool from Bluecurve Inc., to apply stress to our client/server system and measure its performance while under load. We used Dynameasure's OLTP "Order Entry/Mixed Read" test, which contains 13 transactions that perform a combination of SELECT operations against the test SQL database. We chose this test for its flexibility in directing stress at either disk or CPU with minimal tuning required.

intel®

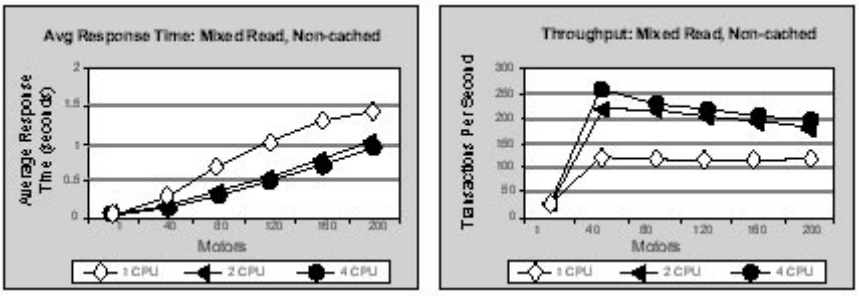

Figure 1: Non-cached performance

Dynameasure presents two main results: transactions per second (TPS) and average response time per transaction (ART). The TPS and ART results from two test runs using 1, 2 and 4 CPUs are shown in Figure 1. In both charts, the horizontal axis indicates increasing load levels, where more motors (user representatives) were added at each step. In the Throughput chart, the vertical axis indicates increasing throughput (higher equates to more work performed). In the Average Response Time chart, the vertical axis represents more time consumed (higher is slower).

The "non-cached" test used to obtain the results in Figure 1 is typical of real world situations in that the data set accessed during the test is too large to fit in memory, thus placing stress on the disk as well. The Throughput chart in Figure 1 indicates that regardless of the number of CPUs, a bottleneck point was reached around 40 motors, since adding more motors beyond 40 had little effect on TPS for each test. Similarly, the ART chart in Figure 1 shows the number of CPUs had little bearing on average response time performance. The most interesting point is that there's relatively little difference between 2 and 4 CPUs. Why didn't a change from 2 to 4 CPUs improve server performance? Chances are, something other than the CPU was the bottleneck—but which part or subsystem was the culprit?

To answer this question, we must look at the utilization of the most likely parts in the server where a bottleneck could develop: CPU and disk. Utilization is a measure of how much a particular system resource—like disk or CPU—is devoted to work. This leads us to a key point about server sizing:

If a resource is too busy, adding more of that resource will probably improve performance.

And a corollary:

If a resource is underutilized, adding more of that resource will probably not improve performance.

Finding a Disk Bottleneck

The Microsoft Windows NT* Performance Monitor that is included with Windows NT is useful for locating the bottleneck point. Figure 2 shows the utilization statistics collected by Performance Monitor during the tests with 1, 2 and 4 CPUs, respectively.

Knowing how to interpret utilization data is key to identifying a bottleneck. Before diving into what the data tell us, we'll use Figure 2 to discuss how to read the charts. There are two lines in the top chart in Figure 2: "CPU 1" marked by diamonds and "Disk" marked by squares. The horizontal axis denotes the load (in motors, equivalent to users). The vertical axis shows how much disk or CPU resource was utilized during a test while we stressed the system. Where a line reaches the top of the chart, the resource it represents is 100% utilized—more to the point, it's the bottleneck. Where a line falls below the top of the chart represents excess capacity—in effect, room to push that resource harder.

Now let's apply this knowledge to the non-cached utilization data in Figure 2. In the top chart (non-cached, 1 CPU), both the disk and CPU appear 100% utilized, as soon as the load reaches 40 motors. Looking at this, one might think this server is balanced, that we have the ideal case where CPU and disk resources are expended evenly. This turns out to be false in this case, though.

We know (from Figure 1) that with 2 CPUs, throughput was significantly better than with 1 CPU. Look at the utilization chart (in Figure 2) for 2 CPUs. Disk is still at 100% but utilization of both CPUs begins to fall off as the load increases. This indicates that disk is the bottleneck with 2 CPUs, and allows us to conclude that the single CPU was a bottleneck in the 1 CPU case. Remember the key principle "adding more of a busy re-source may improve performance"; this is such a situation—we went from 1 to 2 CPUs and obtained substantial performance improvements.

The single CPU case also illustrates a difficulty with bottleneck determinations that involve disk. In these tests, the "disk" was a Windows NT stripe set, made up of 4 drives. Because of the way the Windows NT Performance Monitor reports utilization data, a multiple-disk stripe set may appear busier than it is. When disk and CPU both appear 100% utilized, it's a good idea to vary one resource and measure again to verify. If a resource is too busy, adding more of that resource will probably improve performance.

If a resource is underutilized, adding more of that resource will probably not improve performance.

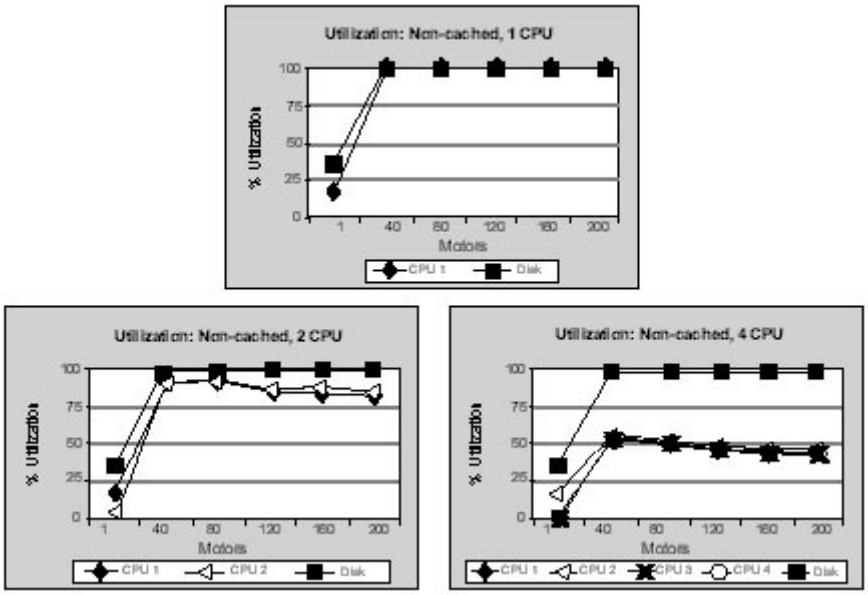

Figure 2: Utilization charts for server CPU and disk resources; non-cached testing

We see further verification of the disk bottleneck in the chart (in Figure 2) for 4 CPUs. The 4 CPUs have plenty of leftover capacity. This explains the lack of significant performance improvement between 2 and 4 CPUs in this test. This is an important point: the capacity of additional CPUs (after 2, in this case) to contribute to improving server throughput or average response time is diminished—indeed almost stopped cold—by the disk.

When, then, will additional CPU capacity improve server performance? Only when other resources are not constrained to the point where they "get in the way," or become a bottleneck. So it stands to reason that if we could get the disk "out of the way," CPU utilization would increase and server performance would improve. In other words, if we improve the disk subsystem, we in effect are trying to eliminate the bottleneck in order to "feed" the CPU with more work. The next test proves this point.

Finding a CPU Bottleneck

Recall that in our previous test we chose a workload that represents a broad mix of transactions to see how the server would behave. Our goal for the next test was to find out how additional CPU capacity can improve throughput and/or response time when there is sufficient disk capacity to "feed" the CPUs. We selected Dynameasure's cache option to ensure that the test data fits into SQL Server's data cache. By fitting the data into cache, we should see less disk utilization and more CPU utilization because the server won't have to visit the disk as frequently to find data. We set Dynameasure's think-time parameter (the time test clients pause between submitting transactions) to 0, to maximize the load.

As we can see in Figure 3, caching the working set had a very significant effect, improving TPS at the peak of the workload by about 96% from 1 to 2 CPUs, and about 90% from 2 to 4 CPUs. The improvement in average response time also supports this finding. At the peak, average response time improved by 50% by adding a 5 second CPU, and again by 50% with the addition of 2 more. Again, let's look at Performance Monitor data to understand why performance improved when we added CPU capacity.

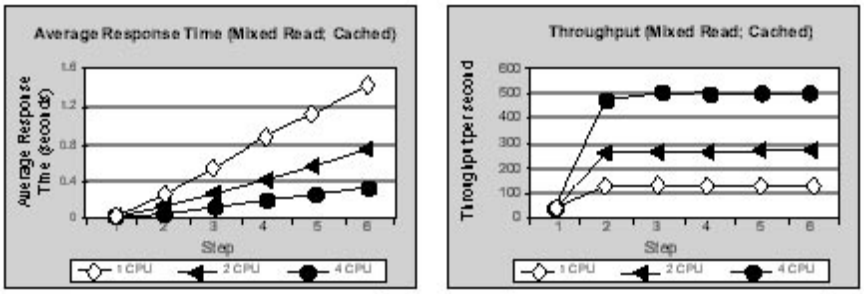

Figure 3: Cached performance

Refer to Figures 2 and 4, which show utilization data for both non-cached and cached tests, respectively. The three charts in Figure 4 reveal that when data is cached, the server's disk is nearly idle, whereas the CPU resource was nearly 100% utilized in 1, 2 and 4 CPU configurations. Our strategy of caching the data worked well to free up the disk in order to stress the CPU resource. When the bottleneck was moved from the disk to the CPU, adding more CPU capacity achieved almost linear scaling on Windows NT.

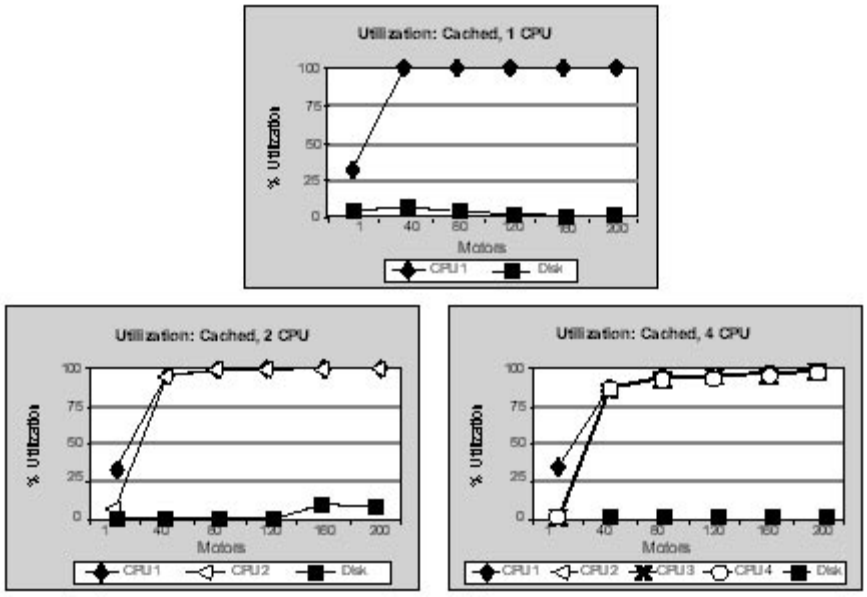

Figure 4: Utilization charts for server CPU and disk resources; cached testing

Conclusion

Striking the right balance among the Dynameasure tests showed that bottlenecks, the part of the computer that is restricting the flow of work, can move among the various parts. Bottlenecks can occur outside of the server, too (clients and networks), but inside the server the CPU and disk subsystems represent two key areas where bottlenecks can occur.

To properly configure a server, we need a picture of its capacity. We can only get a picture of server capacity when the server is being put to work. Knowing how different workloads impact the utilization of various parts tells us how much resource (CPU and disk) is necessary to meet user demands. There are many ways to alter a server's disk subsystem performance, including:

- Number of drives present
- Number of disk controllers
- SCSI, IDE or Fibre Channel data transfer mechanism
- SCSI implementation (SCSI, SCSI-II, SCSI-III, Fast, Wide, Fast-Wide)
- Drive organization (arrays, mirroring, etc.)

The many options can be confusing. Stressing the system as we do here helps quantify real differences in the various choices. Let's look at one of these options—more drives.

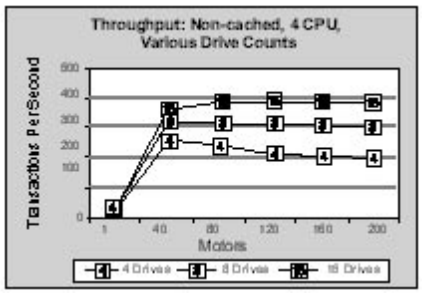

Figure 5: Capacity impact of more drives

We put more disk drives in the server in order to see if performance would improve. The resulting throughput chart (Figure 5) shows that indeed adding more drives dramatically improved throughput in the non-cached test, where the server had previously experienced a disk bottleneck that impeded its ability to utilize the 4 CPUs. The CPU utilization charts (Figure 6) show that the CPUs were fed data more efficiently and thus exhibited higher utilization each time we added more disk drives

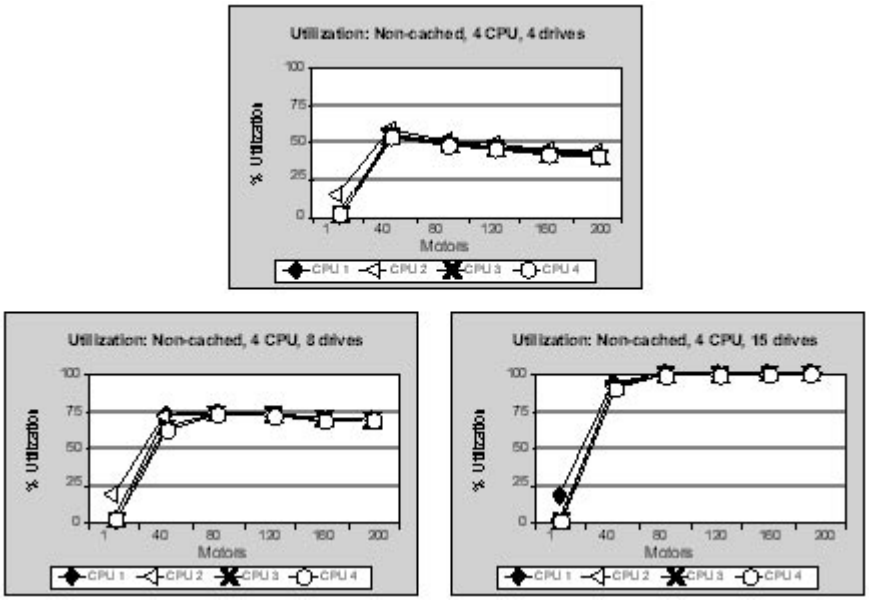

Figure 6: Server CPU utilization as drives are added

What's the bottom line? In the tests where the disk subsystem was the bottleneck, it was unable to effectively feed the 4 CPUs and tap into their capacity to improve performance. When we added more disk drives, we spread the work across more surfaces and made the disk subsystem efficient enough to feed the capacity of the 4 CPUs, and so we saw better performance. So by improving our disk subsystem, we dramatically improved performance by more fully utilizing the potential of multiple CPUs.

By using tools such as Windows NT Performance Monitor and Bluecurve Dynameasure, you can determine how to configure your server to obtain optimal performance for your particular set of circumstances.